

# **MANIPULATING AN ON-SCREEN OBJECT USING ZONES SURROUNDING THE OBJECT**

**Inventor:**

**Sig G. Kupka**

## **Background of the Invention**

### **Cross-Reference to Related Application**

**[0001]** The present application is related to U.S. Utility Patent Application Serial No. \_\_\_\_\_ for "Common On-Screen Zone For Menu Activation And Stroke Input," filed on the same date as the present application. The disclosure of the related application is incorporated herein by reference.

### **Field of the Invention**

**[0002]** The present invention relates generally to user interfaces for managing on-screen objects, and more particularly to a user interface that provides a consistent set of object management zones surrounding an on-screen object.

### **Description of the Background Art**

**[0003]** Existing user interfaces provide many different techniques for moving, altering, controlling, and otherwise manipulating on-screen objects such as windows, images, text blocks, video, and the like. For example, the Windows

XP operating system, available from Microsoft Corporation of Redmond, Washington, provides user interface mechanisms for manipulating various types of on-screen objects. Examples of such user interface mechanisms include:

**[0004]**            - application menus (e.g., click on an object to select it, and select an operation from an application menu);

**[0005]**            - on-screen buttons (e.g., click on an object to select it, and click a button to perform an operation on the object);

**[0006]**            - context-sensitive menus (e.g., right-click on an object and select an operation from a pop-up menu);

**[0007]**            - resize borders or handles (e.g., click-drag a window edge or object handle to resize the window or object); and

**[0008]**            - keyboard commands (e.g., click on an object to select it (or use a keyboard to navigate to an on-screen object), and hit a keystroke to perform an operation on the object).

**[0009]**            One problem with most existing techniques is that there is no consistent user interface paradigm for manipulating objects of different types. For example, the user interface for controlling text objects is significantly different from the user interface for controlling graphic objects. If a user wishes to resize a text object by increasing the text font, he or she performs an entirely different action than he or she would perform for resizing a graphic object. Accordingly, users must learn a variety of different manipulation methods, and know

when to apply which method to which type of object. Often users become disoriented and confused when attempting to control certain types of objects, particularly when the user interface elements for the object being controlled differ from those to which the user has become accustomed.

**[0010]** Furthermore, existing techniques for activating certain object manipulation operations can be cumbersome, difficult to learn, or counterintuitive. For example, specialized on-screen objects, such as objects for representing time periods, often employ different user interface paradigms that may be unfamiliar to users.

**[0011]** In addition, many such techniques do not translate well from one input mechanism to another (such as pen-based, mouse, voice, and keyboard input mechanisms). Users switching from one input mechanism to another must often learn a new object control paradigm in order to use the software effectively.

**[0012]** What is needed, therefore, is a consistent, unified user interface paradigm for providing controls for manipulating on-screen objects, which addresses the limitations of conventional schemes. What is further needed is a user interface paradigm that is extensible and that facilitates ease of use and ease of learning, even when the user is attempting to manipulate different types of objects. What is further needed is a user interface paradigm that is usable with different types of input mechanisms, and that facilitates transitions from one input mechanism to another with minimal disruption, confusion, and re-learning.

## Summary of the Invention

**[0013]** In one aspect of the present invention, each object is surrounded by a number of zones for performing various control operations on the object. The appearance of the zones can vary from one embodiment to another, but in general is kept consistent among various object types within a particular user interface. The zones surrounding an object are referred to collectively as a control region. Each zone of the control region is associated with a series of commands related to a particular type of editing or manipulation function. Displaying the control region around the object enables the user to immediately focus on and identify editing functions that can be performed on the object. Further, because of the proximity of the control region about the object, the user does not need to move the cursor to a distinct menu, tool palette, or other user interface element, and as such, the user can select and execute the desired command more quickly and efficiently.

**[0014]** In one embodiment, one control region is visible at a time, corresponding to the object that is currently active (that is, the object that currently has focus); in another embodiment, control regions for some or all on-screen objects are visible at the same time.

**[0015]** In one embodiment, the control region for an object includes at least nine zones that surround the object itself. The nine zones, including one overlaying the object representation itself, form a three-by-three matrix. In one

embodiment the object representation is located in the same area as the center zone. Thus, eight zones surround the object, forming four corners and four sides, and one zone overlays the object. These zones provide additional control functionality for manipulating the object. In one embodiment, additional zones are provided, including any or all of: a rectangular band-shaped zone located at the outside edge of the center zone, another rectangular band-shaped zone located at the outside edge of the control region, and a rectangular zone located along an edge of the control region. One skilled in the art will recognize that any number of zones may be provided, in any arrangement or layout, without departing from the essential characteristics of the present invention.

**[0016]**           The user specifies commands for manipulating the on-screen object by interacting with the zones within the object's control region. Interaction modes include, for example: pen (touch, press, stroke, or the like); mouse; keyboard; voice; remote controller; and the like. In one embodiment, the user interface of the present invention is capable of receiving user commands in at least two of the above modes, thus allowing a user to interact with the system in a consistent manner using two or more input devices. Commands can be activated in different ways, including menu selection, strokes, clicks, button presses, and the like, performed in the area defined by the corresponding zone. In one embodiment, available commands for interacting with an object including both

geometric editing operations (i.e., those that involve moving, resizing, distorting, and/or rotating), and non-geometric editing operations.

**[0017]** The invention thus provides an easy-to-use, understandable user interface paradigm wherein the control region for each object surrounds the object itself, provides access to all relevant editing operations with respect to the object, and provides multiple input modes that operate consistently, and wherein interactions with zones are consistent from one object type to another.

## **Brief Description of the Drawings**

**[0018]** Fig. 1 depicts an example of a control region for an on-screen object, according to one embodiment.

**[0019]** Fig. 1A depicts an example of a control region for an on-screen object including additional rectangular band-shaped regions.

**[0020]** Fig. 2 depicts an example of a menu associated with a zone, according to one embodiment of the invention.

**[0021]** Figs. 3A and 3B depict examples of command structures for the zones of an object, according to one embodiment.

**[0022]** Figs. 4A and 4B depict examples of a keyboard that includes additional keys corresponding to various zones.

- [0023]** Fig. 5 depicts an example of a remote controller for providing input according to one embodiment of the invention.
- [0024]** Fig. 6 is a screen shot depicting an example of a menu for an input field.
- [0025]** Figs. 7A and 7B depict an example of the operation of the present invention in connection with an input field.
- [0026]** Fig. 8 is a screen shot depicting examples of input objects.
- [0027]** Fig. 9 is a screen shot depicting an example of a menu for a table cell.
- [0028]** Fig. 10 is a screen shot depicting an example of a menu for a table column.
- [0029]** Fig. 11 is a screen shot depicting an example of a menu for a table row.
- [0030]** Fig. 12A depicts an example of a control region for a text object.
- [0031]** Fig. 12B depicts an example of a control region for a text object, the control region including text labels.
- [0032]** Fig. 12C depicts an example of a control region for a text object, the control region including icon labels.
- [0033]** Fig. 12D depicts an example of a control region for a text object, the control region including a pop-up label.

## Detailed Description of a Preferred Embodiment

**[0034]** The following description of system components and operation is merely exemplary of embodiments of the present invention. One skilled in the art will recognize that the various designs, implementations, and techniques described herein may be used alone or in any combination, and that many modifications and equivalent arrangements can be used. Accordingly, the following description is presented for purposes of illustration, and is not intended to limit the invention to the precise forms disclosed.

**[0035]** The invention may be implemented, for example, on a conventional personal computer running any known operating system, such as Microsoft Windows XP available from Microsoft Corporation of Redmond, Washington. As described below, input to the system of the invention can be provided by any of a number of types of devices, including for example: keyboard, mouse, touchscreen, touchpad, trackball, tablet, microphone, remote control device, and/or the like. As will be seen from the following description, the present invention provides techniques for accepting user inputs from various types of input devices and according to various modes in a consistent and intuitive manner.

**[0036]** One skilled in the art will recognize that the user interface techniques of the present invention may be implemented on devices other than personal computers in which a display is provided to the user. For example, such techniques may be implemented on cell phones, consumer electronic devices,

personal digital assistants, office appliances, wall-mounted controllers, automotive control systems, and the like. The description of the invention herein as it might be implemented on a personal computer is not intended to restrict the scope of the present invention or the range of devices on which it can operate.

### **User Interface Layout**

**[0037]** Referring now to Fig. 1, there is shown an example of a user interface 100 according to an embodiment of the present invention. User interface 100 may be displayed, for example, on a display screen of a conventional personal computer. Object 102 is an example of an on-screen object displayed within user interface 100. In the example of Fig. 1, object 102 is a graphic object. However, one skilled in the art will recognize that object 102 can be a text object, or a representation of a video or audio object, or any other type of object that can be displayed or represented within the context of user interface 100.

**[0038]** As shown in Fig. 1, according to one embodiment object 102 is surrounded by various zones 103(A-K), referred to collectively as a control region. Each zone 103 provides a designated type of functionality for controlling one or more aspects of object 102, or otherwise manipulating object 102. Object 102, together with the surrounding control region including zones 103, form a window within user interface 100.

**[0039]** The area of user interface 100 surrounding the window is referred to as a "panel" 104. Any number of windows can be presented within

panel 104 simultaneously, and these windows can overlap one another, or they can be tiled, minimized, or otherwise manipulated according to known user interface techniques as modified and enhanced by the improvements of the present invention. In one embodiment, only one window is active at any given time, and this is the window for the object that has the focus or that is currently active. The user can cause a window (and its object 102) to be active according to well-known techniques such as: moving a cursor to a window and clicking a button; or using a series of keystrokes or voice commands to select and activate a particular window. In one embodiment, zones 103 are displayed only for the currently active window, and are omitted for all other windows. In another embodiment, zones 103 are displayed for all on-screen windows. In addition, in one embodiment panel 104 itself is treated like any other window, so that it can be activated and zones 103 relevant to the control and manipulation of panel 104 can be displayed. Also shown is cursor 110, which is controlled by the user via a mouse or other pointing device, and can be used to select and activate commands and menus as described below.

**[0040]** In the example of Fig. 1, ten zones 103A through 103K are shown, including zone 103J that overlays object 102, and zone 103K that is located outside control region 101. Zone 103K is depicted in the drawings as extending along the bottom of region 101, but in one embodiment includes all areas not within region 101. One skilled in the art will recognize that any number of

zones could be included, and that the particular arrangement of zones 103 shown in Fig. 1 is merely exemplary. For example, as shown in Fig. 1A, additional rectangular band-shaped regions 103L, 103M could be located at the outer edge of zone 103J and at the outer edge of control region 101.

## **Input Mechanisms**

**[0041]** In operation, the device on which the user interface 100 operates detects user commands with respect to the various zones 103 of an onscreen object 102. For example, a user may use a mouse or other input device to move an onscreen cursor 110 to a zone 103. The user may then perform an action, such as clicking a button on a mouse, to activate a command or operation associated with the zone 103. As will be described in more detail below, some zones 103 may have different commands or operations associated with different user actions; for example, a user may click a left button to activate a first command, or a right button to activate a second command. Other possible actions associated with different types of commands include: “stroking” within a zone 103 by holding down a button and moving the cursor 110 a short distance in a particular direction; “shift-clicking” within a zone 103 by holding down a shift key (or other modifier key) while clicking within the zone 103; or double- or triple-clicking within a zone 103.

**[0042]** Any of the above-described input devices can be used for interacting with zones 103. Examples of user action and input mechanisms for interacting with zones 103 include:

**[0043]** - Pen input: Suitable for handwriting input, stroking, gestures, symbols, and abbreviations;

**[0044]** - Mouse or other pointing device: Suitable for clicking, dragging, stroking, and the like;

**[0045]** - Keyboard: Suitable for text entry, navigation among onscreen elements, or single-key activation of commands;

**[0046]** - Voice input: Suitable for activating commands and for inputting text. The user speaks the name of a zone (e.g., "copy") to display a menu for that zone, and then speaks a command from the displayed menu to activate that command.

**[0047]** - Remote control (e.g., RF or IR transmitter) input: Suitable for navigation, data entry, command activation, and possibly cursive writing and stroking.

**[0048]** In one embodiment, the user interface of the present invention is capable of receiving user commands in a consistent manner in two or more of the above modes.

## User Actions

**[0049]** Many different command sets are possible. In one embodiment, the user can perform the following user input actions to activate commands:

**[0050]** - Click standard button (e.g. left-click) while cursor 110 is in zone 103 (or, for pen input, touch zone 103 without pressing pen button): activates a pop-up menu, which may be also called a primary menu, containing commands for zone 103. The displayed pop-up menu shows the relevant commands for the zone 103, and allows the user to navigate and select one of the displayed commands using known menu navigation techniques. In one embodiment, each pop-up menu for each zone 103 is associated with a particular theme or general type of action, or group of associated actions. For example, a zone 103 may have a variety of commands or actions related to font characteristics; another zone 103 may have a variety of commands related to paragraph characteristics, and so forth. In one embodiment, included in the pop-up menu are commands that can be invoked by movement of the input device by a stroke in a predetermined direction (e.g., up, down, left, or right). Fig. 2 illustrates a pop-up menu showing four stroke icons 802, each stroke direction corresponding to a particular command. Also included is a command that corresponds to a double-click for zone 103. One skilled in the art will recognize that the menu can include commands for any other type of action or gesture that can be performed within

zone 103, and may also include commands that cannot be activated directly using an action or a gesture. Referring now to Fig. 2, there is shown an example of such a menu 801, including commands 803 and icons 802 indicating corresponding stroke directions, and commands 803 having icons 802 indicating other actions such as a double-click and an Enter key.

**[0051]**            - Click alternate button (e.g. right-click), or modified button (e.g. shift-click), or click and hold button while cursor 110 is in zone 103 (or, for pen input, either touch zone 103 with pen button pressed, or touch and hold the pen in zone 103): activates an alternate pop-up menu, which may also be called a secondary menu. The alternate pop-up menu operates in a similar manner as the first pop-up menu 801, and also includes icons or indicators showing direct actions that correspond to stroke directions, double-click, and the like, where applicable. In one embodiment, the on-screen cursor 110 changes to indicate that the button has been held long enough to activate the alternate menu.

**[0052]**            - Stroke (i.e., hold down button and move cursor 110 a short distance in a particular direction) starting in zone 103 (or, for pen input, stroke without pressing pen button): performs an action associated with the particular stroke direction. Strokes can be of any type, including straight lines, curves, or more complex shapes or gestures. In one embodiment, as described herein, strokes are associated with particular directions (up, down, left, or right), although one skilled in the art will recognize that other types of strokes may also

be used. As described above, in one embodiment, the pop-up menu 801 for a zone 103 includes each of the stroke directions and indicates what command is associated with each stroke direction. Thus, the user can activate any of these actions by performing the stroke input or by clicking and then selecting the desired item from the pop-up menu 801. Longer strokes can be associated with a click-and-drag action.

**[0053]**            - Double-click (or other action) in zone 103: performs an associated command. As described above, in one embodiment, the pop-up menu 801 for a zone 103 indicates what command is associated with double-clicking (or other actions). Thus, the user can activate the command by double-clicking or by clicking and then selecting the desired item from the pop-up menu 801.

**[0054]**            - Alternate stroke (i.e., hold down alternate button (right button), or modified button (shift-stroke), and move cursor 110 a short distance in a particular direction) starting in zone 103 (or, for pen input, stroke while pressing pen button, or hold pen in place for a moment and then stroke): performs an alternate command associated with the particular stroke direction (up, down, left, or right). As described above, in one embodiment, the alternate pop-up menu for a zone 103 includes each of the stroke directions and indicates what alternate action is associated with each stroke direction. In one embodiment, the on-screen cursor 110 changes when the modifier key is pressed, to remind the user that the alternate command will be performed.

**[0055]** - Alternate double-click (i.e., hold down alternate button (right button), or modified button (shift-stroke), and double-click) in zone 103: performs an associated command. As described above, in one embodiment, the pop-up menu 801 for a zone 103 indicates what command is associated with alternate double-clicking (or other actions). Thus, the user can activate the command by alternate double-clicking or by clicking and then selecting the desired item from the pop-up menu 801.

**[0056]** Alternate menus, stroke commands, double-clicks, and the like can be activated in a number of different ways, depending on the input device. For two-button devices, the alternate button can be used to activate the alternate commands. For one-button devices, the alternate menus and strokes can be activated by pressing and holding the button down for a moment without movement. In one embodiment, the on-screen cursor 110 changes to indicate when the alternate action will be performed. A mechanism for keyboard activation of commands is described in more detail below.

### **Examples of Command Structures**

**[0057]** Referring now to Figs. 3A and 3B, there are shown examples of command structures according to one embodiment, based on the zone layout shown in Fig. 1. Fig. 3A depicts command types corresponding to primary menus, and Fig. 3B depicts command types corresponding to secondary, or alternate, menus.

**[0058]** In general, each zone 103 is associated with a number of actions that are related to a particular function or type, as set forth below. Although these actions are in general consistent among object types, some actions may be object-specific and/or inapplicable to certain types of objects.

**[0059]** In addition to the most commonly used actions that are generally accessible via a primary menu 801, each zone 103 is also associated with an alternate action or set of actions. For example, an alternate menu may be accessed by right-clicking or option-clicking a mouse, or by pressing a shift key prior to pressing a zone 103 activation key. Alternate keystrokes can be accessed by stroking while holding down a right mouse button, or a modifier key such as shift or option. Other techniques may also be used for accessing alternate actions.

**[0060]** The following tables summarize primary command menus for the various zones 103 according to one embodiment. For each type of object listed, a number of available commands are shown; these commands may be available via a primary menu 801 and/or via stroke commands, double-clicks or other actions, as indicated on the table with symbols such as << (left stroke), >> (right stroke), ^ (up stroke), v (down stroke), and : (double-click).

***Upper Left Zone 103A: Start***

Object	Commands
Document	Start (changes top-left corner of document):

	<< Left (unit panel) >> Right (unit panel) /\ Up (unit panel) \ / Down (unit panel) : Reset
<b>While browsing or entering characters:</b> Text File Date Image (entering characters only) Music Video Message	<b>Start (changes first character):</b> << Remove (leading character) >> Add (leading blank) /\ Single quote \ / Cedilla ( , ) : Reset (remove special characters) Parenthesis Bracket Curly bracket
<b>While playing:</b> Music Video Message	<b>Start (changes playing position):</b> << Left (second) >> Right (second) /\ Up (frame 1/30 s) \ / Down (frame 1/30 s) : Reset (start to 0:00:00)
Image	<b>Start (changes top-left corner cropping):</b> << Left (pixel) >> Right (pixel) /\ Up (pixel) \ / Down (pixel) : Reset
Shape	<b>Start (changes top-left extent of Shape):</b> << Left (pixel) >> Right (pixel) /\ Up (pixel) \ / Down (pixel) : Reset (align to grid)
Overlay objects (excludes: Document, Column, Row, Cell)	<b>Start (changes top-left extent of Object):</b> << Left (pixel) >> Right (pixel) /\ Up (pixel) \ / Down (pixel) : Reset (align to grid)
Paragraph List Item in List Table	<b>Start (changes top-left margin):</b> << Left (character) >> Right (character) /\ Up (0.2 line) \ / Down (0.2 line) : Reset
Date	<b>Start (Date object):</b> << Left (back 1 month) >> Right (forward 1 month) /\ Up (forward 1 day)

	\ / Down (back 1 day) : Today Yesterday Tomorrow
Number	Start (numeric value): << Left (much smaller) >> Right (much larger) /\ Up (larger) \ / Down (smaller) : Round (zero last non-zero digit)
Time	Start (start time): << Left (qtr.hour) >> Right (qtr.hour)) /\ Up (minute) \ / Down (minute)
Column Row Cell	Not Available

***Upper Middle Zone 103B: Insert***

Object	Commands
<b>While browsing:</b> Document Text File Date Image Music Video Message Shape Number Time Paragraph List Item in List Table Row Column Cell	<b>Insert:</b> << Insert contents of clipboard before current object >> Insert contents of clipboard after current object \ / Add current object to clipboard /\ Replace clipboard with copy of current object : Save (link names for later insert) Link name 1 Link name 2 .... Link name n
<b>While entering characters:</b> Text File Image Music Video Message	<b>Insert:</b> << Insert contents of clipboard before cursor position >> Insert contents of clipboard after cursor position /\ Replace clipboard with text of current object \ / Add text of current object to clip-

	board  <b>Insert (Clipboard with 1 objects):</b> << Insert contents of clipboard before current selection >> Insert contents of clipboard after current selection /\ Replace clipboard with current selection \ / Add current selection to clipboard
--	--

***Upper Right Zone 103C: Delete***

Object	Commands
<b>Document:</b>	<b>Delete (entire document):</b> /\ Replace (Clipboard with document) \ / Add (to Clipboard) : This
<b>While browsing:</b> Text File Date Image Music Video Message Shape Number Time	<b>Delete (objects):</b> << Before (this object) >> After (this object) /\ Replace (Clipboard with this object) \ / Add (this object to Clipboard) : This (object)
<b>While entering characters:</b> Text File Image Music Video Message	<b>Delete (characters):</b> << Before (cursor) >> After (cursor) /\ Replace (Clipboard with text of object) \ / Add (text of object to Clipboard) : This (adjacent word) <b>Delete (characters):</b> << Before (selection) >> After (selection) /\ Replace (Clipboard with selection) \ / Add (selection to Clipboard) : This (selection)
<b>Paragraph :</b> List Item in List Table Row Column Cell	<b>Delete (objects):</b> << Before (this object) >> After (this object) /\ Replace (Clipboard with this object) \ / Add (this object to Clipboard) : This (object) Content (keep this object)

### *Middle Left Zone 103D: Format*

Object	Commands
<b>Document :</b> Text File Date Image Music Video Message Number Time Paragraph List Item in List Table Row Column Cell	<b>Format:</b> << Cross out (Strike-through) >> Italic /\ Bold \/ Underline : Reset (normal) Save (format) Apply (saved format) Apply current format to preceding object Apply current format to successive object
<b>Shape :</b> Line Arc Scribble	<b>Format:</b> << Starting arrow >> Ending arrow Save (format) Apply (saved format) Apply current format to preceding object Apply current format to successive object
<b>Shape :</b> Rectangle Oval Point	Not available

### *Middle Right Zone 103E: Create*

Object	Commands
<b>While browsing:</b> Document Text File Date Image Music Video Message Shape Number	<b>Create (new object):</b> << Before (same) >> After (same) /\ Parent (same as parent) \/ Child (same as last child) : Text Time Number Scribble (Shape) Line (Shape) Rectangle (Shape)

Time Paragraph List Item in List Table Row Column Cell	Arc (Shape) Oval (Shape) Point (Shape) Paragraph Group List Table Date File Image Message Music Video
While entering characters: Text File Image Music Video Message	Create (abbreviation): << Collapsed (long -> short) >> Expanded (short -> long) /\ Deletion (from abbreviation list) \/ Addition (short   long) : All (expand all abbreviations) Long 1 (short 1) Long 2 (short 2) .... Long n (short n)
Overlay objects (excludes: Document, Column, Row, Cell)	Create (new object): << Left (same to left) >> Right (same to right) /\ Up (same above) \/ Down (same below) : Text Time Number Scribble (Shape) Line (Shape) Rectangle (Shape) Arc (Shape) Oval (Shape) Point (Shape) Paragraph Group List Table Date File Image Message Music Video

**Lower Left Zone 103F: Change To**

Object	Commands
Document	<b>Change (Bookmark) to:</b> << Remove >> Add /\ Previous (first) \ / Next (last) : Reset (load Resource file) Bookmark 1 Bookmark 2 .... Bookmark n
<b>While browsing or entering characters:</b> Text File Image (entering characters only) Music Video Message	<b>Change (characters) to:</b> << First letter CAP >> Title case /\ Uppercase \ / Lowercase : Reset (automatic CAPS)
<b>While playing:</b> Music Video Message	<b>Change (playing position) to:</b> << Fast reverse (15 seconds) >> Fast forward (15 second) /\ Previous (clip) \ / Next (clip) : Reset (to start of clip) Random (order of play) Sequential (order of play)
Date	<b>Change (Date) to:</b> << Lower (year) >> Higher (year) /\ Previous (type) \ / Next (type) : Reset (Aug 10, 2003) Month day (August 10) Long (August 10, 2003) Day of week (Sunday August 10, 2003) Concise (8/10/03) Minimum (8/10)
Paragraph List Item in List Table	<b>Change (Paragraph) to:</b> << Left (objects flow to right) >> Right (objects flow to left) : Reset (no flow) Title case (override all) Uppercase (override all) Lowercase (override all) None (remove CAPS override)
Row Column	<b>Change to:</b> Title case (override all)

<b>Cell</b>	Uppercase (override all) Lowercase (override all) None (remove CAPS override)
<b>Number</b>	<b>Change (Number) to:</b> << Smaller (divide by 10) >> Larger (multiply by 10) /\ Previous (type) \ / Next (type) : Reset Normal (Integer or decimal) Scientific notation Percent Dollars
<b>Time</b>	<b>Change (alarm start time) to:</b> << Left (qtr.hour) >> Right (qtr.hour)) /\ Up (minute) \ / Down (minute) : Reset (remove alarm)
<b>Shape Image</b>	<b>Change (rotation)</b>

***Lower Middle Zone 103G: Shift***

<b>Object</b>	<b>Commands</b>
<b>While browsing:</b> Document Paragraph List Item in List Table Column Row Cell	<b>Shift:</b> << First (child object) >> Last (child object) /\ Parent \ / Child : Play mode
<b>While entering characters:</b> Text File Image Music Video Message	<b>Shift:</b> << First (char. text entry) >> Last (char. text entry) /\ Parent : Play mode
<b>While playing:</b> Music Video Message	<b>Shift (playing position):</b> << Left (second) >> Right (second) /\ Up (frame 1/30 s) \ / Down (frame 1/30 s) : Reset (to start of clip)
<b>Shape</b>	<b>Shift (opacity=36%):</b> << Less (opaque)

	>> More (opaque) /\ Transparent \ / Opaque
Number Time	Not Available

***Lower Right Zone 103H: End***

Object	Commands
Document	<b>End (changes bottom-right corner of document):</b> << Left (unit panel) >> Right (unit panel) /\ Up (unit panel) \ / Down (unit panel) : Reset
<b>While browsing or entering characters:</b> Text File Date Image (entering text only) Music Video Message	<b>End (changes terminating character):</b> << Remove >> Question mark /\ Exclamation point \ / Semicolon : Period Single quote (surrounding object) Double quote (surrounding object) Parenthesis (surrounding object) Bracket (surrounding object) Curly bracket (surrounding object) Copyright ( © ) Registered (trademark ® )
<b>While playing:</b> Music Video Message	<b>End (changes playing position):</b> << Left (second) >> Right (second) /\ Up (frame 1/30 s) \ / Down (frame 1/30 s) : Reset (to end of clip)
Image	<b>End (changes bottom-right corner cropping):</b> << Left (pixel) >> Right (pixel) /\ Up (pixel) \ / Down (pixel) : Reset (remove cropping)
Shape	<b>End (changes bottom-right extent of Shape):</b> << Left (pixel) >> Right (pixel) /\ Up (pixel) \ / Down (pixel) : Reset (align to grid)

Overlay objects (excludes: Document, Column, Row, Cell)	<b>End (changes bottom-right extent of Object):</b> << Left (pixel) >> Right (pixel) /\ Up (pixel) \ / Down (pixel) : Reset (align to grid)
Paragraph List Item in List Table	<b>End (changes bottom-right margins):</b> << Left (character) >> Right (character) /\ Up (0.2 line) \ / Down (0.2 line) : Reset
Column	<b>End (changes column width, absolute):</b> << Left (0.1 character) >> Right (0.1 character) : Reset (to automatic width) Convert (to % width) Before (apply same width) After (apply same width) <b>End (changes column width, relative):</b> << Left (0.1%) >> Right (0.1%) : Reset (to automatic width) Convert (to character width) Before (apply same width) After (apply same width)
Row	<b>End (changes Row height, absolute):</b> /\ Up (0.1 line) \ / Down (0.1 line) : Reset (to automatic height) Convert (to % height) Before (apply same height) After (apply same height) <b>End (changes Row height, relative):</b> /\ Up (0.1%) \ / Down (0.1%) : Reset (to automatic height) Convert (to character height) Before (apply same height) After (apply same height)
Cell	<b>End (changes number of rows/columns spanned by cell):</b> << Left (reduce cell span by one column) >> Right (extend cell span by one column) /\ Up (reduce cell span by one row) \ / Down (extend cell span by one row) : Reset (to 1 column and 1 row)
Number	<b>End (changes width):</b>

	<< Left (1 character narrower) >> Right (1 character wider) /\ Reset (to standard width)
Time	<b>End (changes duration):</b> << Left (qtr. hour less) >> Right (qtr. hour more) /\ Up (1 minute more) \ / Down (1 minute less)

***Center Zone 103J: Drag***

Object	Commands
Document	<b>Drag (document):</b> << Left (unit panel) >> Right (unit panel) /\ Up (unit panel) \ / Down (unit panel) : Enter (activate child object)
<b>While entering characters:</b> Text File Date Image Music Video Message	Select characters or insertion point
Text (while browsing)	<b>Drag:</b> << Before >> After : Enter (start text entry)
<b>While browsing:</b> File Date Image Music Video Message	<b>Drag:</b> << Before >> After : Enter (load object)
<b>While playing:</b> Image Music Video Message	<b>Drag (Play mode):</b> : Exit
Absolute positioned objects (excl. Document, Item in List, Column, Row, Cell)	<b>Drag (Overlay object):</b> << Left (pixels) >> Right (pixels) /\ Up (pixels) \ / Down (pixels) : Enter (activate child object)
Paragraph	<b>Drag:</b>

Item in List Cell	<< Before >> After : Enter (activate child object) Drag (empty object): : Enter (create Text object)
Table Row Column List	Drag: << Before >> After : Enter (activate child object)
Number Time Shape (In-flow)	Drag: << Before >> After

***Outside Zone 103K: Go To***

**[0061]** If the user clicks on a control region 101 outside the currently active control region 101, the object corresponding to the selected control region 101 is made active.

**[0062]** In one embodiment, whenever the user clicks at a location not within a control region 101 of an existing object, a new copy of a “resource” document is loaded into the empty space. The resource document is the highest-level object in the object hierarchy.

**[0063]** If the user drags the cursor 110 across a series of objects, these objects are highlighted. When the button is released or the pen is lifted, the most recently highlighted object is activated and a new control region 101 appears surrounding this object. If the user moves the cursor 110 beyond the limits of the active object, the contents of the object are automatically scrolled to make additional objects accessible for activation.

**[0064]** Additional navigation among objects is available by activating a menu corresponding to outside zone 103K, or by performing a stroke or other action as listed below, while the cursor 110 is located in outside zone 103K.

Go to:  
First (sibling)  
Last (sibling)  
<< Previous  
>> Next  
/\ Parent  
\ / Child  
: This (Document)  
Resource 1  
Resource 2  
....  
Resource n

**[0065]** Resources 1 through n are defined in the “Resource” document and specify all major groupings of user information, hardware devices, and network resources, as described in more detail below. Typical resource definitions include:

Preferences

House

Media

Address

Name

EMail

Date

Subject

Phone

Web

Offline Storage

Printers

**[0066]** The following tables summarize alternate command menus for the various zones 103 according to one embodiment. For each type of object listed, a number of available commands are shown; these commands may be available via an alternate menu (activated, for example, by right-clicking) and/or via alternate stroke commands, alternate double-clicks or other actions, as indicated on the table with symbols such as << (left stroke), >> (right stroke), ^ (up stroke), v (down stroke), and : (double-click).

***Upper Left Zone 103A (Alternate): Fill Color***

**[0067]** Fill color can be applied to any objects except for those that have no interior space (such as lines, arcs, and points). Activating the fill color menu causes a palette to appear, from which the user can select the desired color.

***Upper Middle Zone 103B (Alternate): Zoom***

Object	Commands
Document	Zoom:
Text	<< Smaller
File	>> Larger
Date	/\ Previous (font)
Music	\ / Next (font)
Message	

<b>Number</b> <b>Time</b> <b>Paragraph</b> <b>List</b> <b>Item in List</b> <b>Table</b> <b>Row</b> <b>Column</b> <b>Cell</b>	: Reset (standard size and font) Serif Cursive Monospace
<b>Image</b> <b>Video</b>	<b>Zoom:</b> << Smaller >> Larger /\ Thumbnail : Reset (revert to original size)
<b>Shape:</b> Rectangle Oval Line Arc Point	<b>Zoom:</b> << Smaller >> Larger
<b>Shape:</b> Scribble	<b>Zoom:</b> << Smaller >> Larger : Reset (revert to original size)

***Upper Right Zone 103C (Alternate): Undo***

**[0068]** Alternate-clicking within zone 103C activates an undo command. The user can also activate the alternate menu to navigate within the history of commands, as follows:

**Undo (2 History objects):**  
<< Older (show History)  
>> Newer (show History)  
\ / Show all  
/\ All  
: This (most recent History)

***Middle Left Zone 103D (Alternate): Text Color***

**[0069]** Text color can be applied to any object that includes text. Activating the text color menu causes a palette to appear, from which the user can select the desired color.

*Middle Right Zone 103E (Alternate): Transform To*

Object	Commands
Document	Transform to: Protected Unprotected Cleaned up Original version Save
Text	Transform to: << Combine >> Split File Date
File	Transform to: >> Split \\ / Displayed content Text
Date	Transform to: >> Split \\ / Displayed content Text
Image Music Video Message	Transform to: >> Split \\ / Displayed content
Shape- Line	Transform to: Rectangle Oval
Shape- Arc	Transform to: Line Rectangle Oval
Shape- Scribble	Transform to: Line Rectangle Oval
Shape- Rectangle	Transform to: Line Oval
Shape- Oval	Transform to: Line Rectangle
Shape- Point	Transform to: Oval
Paragraph List Item in List Table	Transform to: << Combine >> Split \\ / Displayed content

	List
Column Row	Transform to: << Combine >> Split
Cell	Transform to: << Combine >> Split List
Number Time	Transform to: >> Split Text

***Lower Left Zone 103F (Alternate): Border Color***

**[0070]** Border color can be applied to any object potentially having a border. This includes, for example, images, numbers, times, paragraphs, lists, items in lists, tables, and cells. Activating the border color menu causes a palette to appear, from which the user can select the desired color.

***Lower Middle Zone 103G (Alternate): Align***

Object	Commands
Document Music Video Message Image	Not available
Text File Date	Align (text vertically): /\ Up (superscript) \ / Down (subscript) : Reset (normal) Overlay (NOT in-flow positioning) Before (same as this) After (same as this)
Shape	Align (to grid): << Left >> Right /\ Up (larger grid size) \ / Down (smaller grid size) : Reset (normal) In-flow (Not overlay positioning) Before (same as this) After (same as this)

Overlay objects (excludes: Document, Column, Row, Cell)	<b>Align (to grid):</b> << Left >> Right /\ Up (larger grid size) \\ Down (smaller grid size) : Reset (normal) In-flow (Not overlay positioning) Before (same as this) After (same as this)
Paragraph List Item in List Table	<b>Align (contained objects horizontally):</b> << Left >> Right /\ Justify text (and left align) \\ Center : Reset (normal) Overlay (NOT in-flow positioning) Before (same as this) After (same as this)
Row Column Cell	<b>Align (cell content horizontal &amp; vertical):</b> << Left >> Right /\ Top \\ Bottom : Reset (normal) Center (horizontal & vertical) Horizontal center Vertical center Before (same as this) After (same as this)
Number	<b>Align (decimal point position):</b> << Left >> Right : Reset (normal) Overlay (NOT in-flow positioning) Before (same as this) After (same as this)
Time	<b>Align (object):</b> Overlay (NOT in-flow positioning) Before (same as this) After (same as this)

***Lower Right Zone 103H (Alternate): Border***

Object	Commands
Document Text File Date	Not available

<b>Music</b> <b>Video</b> <b>Message</b>	
<b>Image</b> <b>Number</b> <b>Time</b> <b>Paragraph</b> <b>List</b> <b>Item in List</b> <b>Table</b> <b>Cell</b>	<b>Border (surrounding object):</b> << Thinner >> Thicker /\ Previous (border type) \\/ Next (border type) : Reset (standard border) None Ridge Solid Double Groove Inset (current) Outset Dashed Dotted
<b>Row</b>	<b>Border (between rows):</b> \\/ Add : Reset (between columns & rows) <b>Border (between rows):</b> /\ Remove : Reset (between columns & rows)
<b>Column</b>	<b>Border (between columns):</b> \\/ Add : Reset (between columns & rows) <b>Border (between columns):</b> /\ Remove : Reset (between columns & rows)
<b>Shape</b>	<b>Border (line type &amp; width=4):</b> << Thinner >> Thicker /\ Previous (border type) \\/ Next (border type) : Reset (single, solid, width=2) Single (current) Double Triple Solid (current) Dotted Dashed Dot Dash

*Center Zone 103J (Alternate): Copy*

**[0071]** When the user alternate-clicks within zone 103J, a copy of the active object is copied by moving the outline of the active object to another destination. The destination object can be in another document.

**[0072]** The user can activate a menu for positioning and manipulating the copy of the currently active object. The menu includes the following commands (some or all of which are also available via strokes):

Copy (move outline to destination):  
/\ Up (pixels)  
\ / Down (pixels)  
<< Left (pixels)  
>> Right (pixels)  
: Exit (activate this document)

*Outside Zone 103K (Alternate): Scroll*

**[0073]** Scroll commands are accessed by alternate-clicking outside control region 101. As long as the button is held or the pen stays in contact with the display, the currently active object is directly scrolled using the pen or mouse movement. In one embodiment, as the scrolling limits of the object are reached, the object's outer edge (corresponding to the direction of scrolling) changes color.

**[0074]** In another embodiment, an alternate-click activates a scroll menu containing the following commands:

Scroll (document):  
<< Left (half page)  
>> Right (half page)  
/\ Up (half page)

\ / Down (half page)  
: Reset (top or bottom)

**[0075]** As with the other menus described above, commands can also be directly activated by performing the stroke, double-click, or other action indicated.

**[0076]** Wherever possible, each zone provides unified, consistent access to a group of commands that are related to a particular theme. The input actions of clicking, click-and-hold, or stroking provide various ways to access commands within the group.

**[0077]** For example, zone 103A operates as follows in connection with objects of a time-varying nature (such as audio, video, animations, and the like). Zone 103A provides commands for changing the start of the data object (for example, if the data object is a video stream, zone 103A allows manipulation of the time code at which the video stream begins). As described in the table above, stroking in zone 103A changes the start point by an amount and direction that depend on the direction of the stroke. For example, vertical strokes would cause coarse adjustments to the start point to be made (changing the minutes), while horizontal strokes would cause fine adjustments to be made (changing the seconds). Alternatively, vertical strokes might change the value by a one-hour increment (or some other relatively coarse increment), while horizontal strokes might change the value by a one-minute increment (or some other relatively fine

increment). In general, strokes in the up and left directions move the start point earlier, while strokes in the down and right directions move the start point later. The particular increments and directions of movement can be preset, or can be user-configurable via a preferences or options screen.

### **Example: Input Field**

**[0078]** Referring now to Figs. 7A and 7B, there is shown an example of the operation of the present invention in connection with a particular type of object: an input field 700. Fig. 7A depicts input field 700 before control region 101 has been made active (for example, before input field 700 has focus or has been selected). The example input field 700 of Fig. 7A indicates a time period (9:00 – 9:45), as may be included for example in a calendaring program for maintaining information about appointments and reminders. Input field 700 is an object that has two primary purposes: 1) to display a representation of a time period; and 2) to accept user input for changing the time period and/or the manner in which it is displayed.

**[0079]** In one embodiment, input field 700 is made active (or has “focus”) when a user clicks or taps within its bounds or otherwise selects the field. As shown in Fig. 7B, control region 101, including zones 103, then becomes visible. The example of Fig. 7B illustrates how zones 103, described in general terms above, provide specific functionality for facilitating user input to change the value and format for the time period represented by input field 700. Specifically:

**[0080]** - Zone 103A allows the user to change the start of the time period. As described above, vertical strokes indicate coarse changes, while horizontal strokes indicate fine changes. In one embodiment, the time period adjustment is repeated if the user holds the stylus or other pointing device in position at the end of the stroke; in another embodiment, the time period adjustment is repeated if the user continues to stroke in the same direction (if desired, a wait period can be implemented before the repetition commences, as is conventionally done in auto-repeat modes for keyboards). The time period adjustment repetition can be set to accelerate if the user continues to hold for a predetermined period of time. In one embodiment, the user can “back up” if the time period adjustment goes to far. The display of input field 700 is continually adjusted so that the user can see the time period adjustments in real-time and release or backtrack at precisely the correct moment.

**[0081]** - Zone 103H allows the user to change the end point of the time period. End point adjustment is performed using the same coarse/fine adjustment paradigm described above in connection with zone 103A.

**[0082]** - Zones 103B, 103C, 103D, 103E, 103F, 103G, and 103J provide commands as described above.

**[0083]** In one embodiment, the duration of the event remains constant when the start point is changed (so that the end point changes by the same amount as the start point). In another embodiment, the end point is held con-

stant when the start point is changes (so that the duration changes). In yet another embodiment, the user can shift-click or indicate by a modifier key which of these two modes of operation is desired; alternatively, the mode can be set via a preferences or options screen.

### **Example: Tables and Cells**

**[0084]** The present invention provides a consistent interface for manipulating the characteristics of table objects and cell objects. Sizes, such as row heights, column widths, and overall table sizes, can be specified in absolute terms or as a percentage of a containing object. For example, a cell width can be configured to be 25% of the width of the containing table; if the table width is increased, the cell width increases proportionately. The user interface allows for selection and operation on individual cells, rows, columns, or the entire table. In one embodiment, for example, clicking on a corner of the table selects the entire table; clicking on the left or right edge selects the table row at the clicked location; clicking on the top or bottom edge selects the table column at the clicked location; and clicking on a cell selects the individual cell. In one embodiment, the control region is presented for the selected cell, column, row, or table, so that the user can then activate commands within the various zones of the active control region.

## **Keyboard Input**

**[0085]** In addition to the input modes described above, the user can also activate commands via a standard keyboard such as a QWERTY keyboard. In one embodiment, each zone 103 is associated with a keystroke; striking the corresponding key causes the menu 801 for that zone 103 to be displayed. Stroke commands, double-click commands, and the like can also be associated with keystrokes, so that the user can activate the command by striking the corresponding key (accompanied, perhaps, by a Control key, an Alt key, or some other modifier key). Letters, or other indicators, can be displayed within zones 103 to show which keystroke corresponds to each zone 103. Keystrokes for zones 103 can be configured in a mnemonically suitable manner (for example, Ctrl-G corresponds to Go To, while Ctrl-C corresponds to copy).

**[0086]** Referring now to Figs. 4A and 4B, there is shown an example of a keyboard 400 that includes additional keys 401 corresponding to various zones 103. Keys 401 are positioned around the perimeter of a standard QWERTY keyboard 403, with each key 401 in a location that maps to the corresponding zone 103. For example, key 401A, which activates the start menu 801, is located at the top left corner of keyboard 400, a location that corresponds to the location of zone 103A that also activates the start menu 801. Similar correspondences apply for keys 401B through 401H. Alternate menus can be activated, for example, by holding down shift key 404 while pressing the desired zone key 401.

**[0087]** In the example of Fig. 4A, keyboard 400 also includes keys 402U, 402D, 402L, 402R that, when pressed, perform stroke actions corresponding to up, down, left, and right strokes. A user can press a key 401 followed by a stroke key 402 to perform an action in the zone 103 corresponding to the pressed key 401. In one embodiment, when the user presses key 401, the system waits a short, predetermined period of time (such as ¼ second) before presenting the menu associated with key 401. If, during that waiting period, the user presses one of keys 402U, 402D, 402L, 402R, or if he or she presses key 401 again, the command associated with the specified stroke or double-click is immediately performed and the menu is not displayed. If, during the waiting period, the user presses an alphabetic key indicating a desired command from the menu, the command is executed immediately, again without the menu itself being displayed. If the user performs neither of these actions during the waiting period, the system proceeds with display of the menu. In this manner, the system accommodates expert users (who are familiar with the command structure and do not need to see the menus before activating commands within them), and novices (who wish to see the available commands before activating one).

**[0088]** Alternatively, an alternate stroke action can be activated by pressing shift plus a key 401 followed by a stroke key 402. Stroke keys 402 can be configured to auto-repeat when held down for a predetermined length of time.

In one embodiment, pressing the Enter key is equivalent to double-clicking on the zone 103 corresponding to the most recently pressed key 401.

**[0089]** In the example of Fig. 4B, keyboard 400 includes touchpad 405 for performing stroke actions; these replace stroke keys 402 of Fig. 4A. Moving a finger across touchpad 405 is equivalent to a pen stroke or mouse movement, a tap is equivalent to a click, and a double-tap is equivalent to a double-click. A user can press a key 401 and stroke or tap touchpad 405 to perform a stroke action in the zone 103 corresponding to the pressed key 401. Alternatively, stroke action can be activated by pressing shift plus a key 401 and then stroking or tapping touchpad 405. For example, to capitalize first letters of each word in the active object, a user presses Change To button 401F and then performs a right stroke on touchpad 405.

**[0090]** One skilled in the art will recognize that keys 401 and 402 may be implemented as standard keys or buttons, or touch-sensitive regions surrounding QWERTY keyboard 403, or by some other technique. Alternatively, a standard keyboard can be used, wherein some keys are reassigned, either temporarily or permanently, to perform commands analogous to zones 103. In one embodiment, keys are reassigned according to their physical locations on the keyboard. For example, a standard "Esc" key accesses the Start command corresponding to zone 401A; a Tab key accesses the Format command corresponding to zone 401D; standard arrow keys can activate strokes, and the like. Alternate

menus can be activated, for example, by holding down shift key 404 while pressing the desired zone key 401.

**[0091]** In another embodiment, a standard keyboard can be used, and menus 801 and commands can be chosen via keystrokes that correspond to the first letter (or some other letter) of the menu 801 or command. Arrow keys can be used to navigate within a menu 801 once it has been displayed. An enter key can be used to activate the currently highlighted menu item.

**[0092]** Using the above techniques, the present invention can be implemented in a manner that allows all input to be provided on a keyboard without requiring pen, mouse, or other pointing device.

### **Voice Input**

**[0093]** In one embodiment, the user interface of the present invention can be combined with existing voice-activated user interface systems. The user can activate a command by speaking a word corresponding to the desired command. Voice recognition software, as is well known in the art, interprets signals from the microphone to determine what words were spoken by the user. The command corresponding to the spoken word is then performed.

**[0094]** In one embodiment, the user activates a voice command by first speaking one of the words corresponding to a zone 103 of a control region. The command menu 801 for that zone 103 is displayed, and the user can activate a command within the menu 801 by speaking the word corresponding to the de-

sired command. Alternatively, if the user wishes to bypass the menu display, he or she can simply speak the word for a zone 103 followed immediately by the name of the command within the menu 801 for that zone 103. For example, a user can say "Format"; the format menu 801 is then displayed; and the user can say "Bold" to choose the Bold command from the format menu 801. Alternatively, the user can simply say "Format Bold" to activate the Bold command.

**[0095]** In one embodiment, the system of the present invention also accepts voice input for dictation and transcription. Thus, the user can enter text by speaking into the microphone; the system interprets the spoken words as text input. Such functionality is implemented according to well known voice input techniques.

### **Remote Control Input**

**[0096]** In one embodiment, the system of the present invention accepts input provided via a remote controller. Referring now to Fig. 5, there is shown an example of a remote controller 500 according to one embodiment of the invention. Remote controller 500 transmits signals, such as infrared or RF signals, to a device (not shown) supporting the present invention via transmitter 501 according to well-known techniques.

**[0097]** In one embodiment, remote controller 500 includes touchpad 502 that is subdivided into a number of zones 503A through 503K. Zones 503A through 503H are arranged in a similar manner to zones 103A through 103K of

the on-screen interface depicted in Figs. 3A and 3B. Touching or stroking any of zones 503A through 503K effects the same result as touching or stroking corresponding zone 103A through 103K as described above. Touching a zone 503A through 503K and holding the finger for a moment prior to moving activates the zone's alternate (or secondary) command, as described above. In addition, touchpad 502 includes zone 503L which controls volume (for example, by stroking up or down to increase or decrease volume, or tapping to mute and unmute). The remote controller 500 generates an appropriate signal based on the touched/stroked zone 503, and transmits the signal to the device. The device has a signal receiver (e.g., IR receiver or RF receiver), which receives the signals, converts it to the appropriate format and semantics if needed by the operating system of the device, which provides the signal to an application having the user interface 100.

**[0098]** In another embodiment, remote controller 500 includes a number of buttons (not shown) that replace touchpad 502 and perform analogous functionality to zones 503 described above. Alternatively, one skilled in the art will recognize that other input knobs, buttons, wheels, and the like, may be provided instead of or in addition to those described herein. In addition, one skilled in the art will recognize that the particular arrangement of zones 503 in Fig. 5 is merely exemplary.

## Screen Shot Examples

**[0099]** Referring now to Fig. 6, there is shown a screen shot depicting an example of a menu 801 for an input field 700. Menu 801 is activated by clicking on the lower right zone of control region 101. Input field 700 contains the value "05:30p - 05:45p." Menu 801 includes commands 803 for changing the end time by various increments. Icons 802 indicate stroke directions corresponding to commands 803. In the example, horizontal strokes are used for coarse adjustments (by quarter-hour increments), and vertical strokes are used for fine adjustments (by one-minute increments).

**[0100]** Referring now to Fig. 8, there is shown a screen shot depicting a window 800 including examples of input fields 700, 700A. Various types of numeric data are shown in input fields 700, 700A. The techniques of the present invention can be used to adjust or manipulate the values and formats for all of the examples shown, among others. One input field 700A currently has focus, and the user has activated menu 801 for input field 700A by clicking on the lower right zone 103H of control region 101. As in the example of Fig. 6, input field 700A contains the value "05:30p - 05:45p." Menu 801 includes commands 803 for changing the end time by various increments. Icons 802 indicate stroke directions corresponding to commands 803. Again, horizontal strokes are used for coarse adjustments (by quarter-hour increments), and vertical strokes are used for fine adjustments (by one-minute increments).

**[0101]** Referring now to Fig. 9, there is shown a screen shot depicting an example of a menu 801 for a table cell 902A. Table 901 includes a number of cells 902, 902A. Cell 902A (which spans two columns of table 901) currently has focus, and the user has activated menu 801 by clicking on the lower right zone 103H of control region 101. Menu 801 includes commands 803 for reducing or extending cell 902A span in the horizontal and vertical directions, as well as a command 803 for resetting cell 902A span to one column and one row. Icons 802 indicate stroke directions corresponding to commands 803. One icon 802 is a ":" symbol, indicating that the corresponding action for direct activation of the command is a double-click.

**[0102]** Referring now to Fig. 10, there is shown a screen shot depicting an example of a menu 801 for a table column. Table 901 includes a number of cells 902, 902B. Cells 902B collectively form a column that currently has focus. The user has activated menu 801 by clicking on the lower right zone 103H of control region 101. Menu 801 includes commands 803 for changing the width of the column and for resetting the width to an automatic width. Icons 802 indicate stroke directions corresponding to commands 803. One icon 802 is a ":" symbol, indicating that the corresponding action for direct activation of the command is a double-click. Menu 801 also includes additional commands 803A that are not directly activatable via a stroke or action.

**[0103]** Referring now to Fig. 11, there is shown a screen shot depicting an example of a menu 801 for a table row. Table 901 includes a number of cells 902, 902C. Cells 902C collectively form a row that currently has focus. The user has activated menu 801 by clicking on the lower right zone 103H of control region 101. Menu 801 includes commands 803 for changing the height of the row and for resetting the height to an automatic height. Icons 802 indicate stroke directions corresponding to commands 803. One icon 802 is a “:” symbol, indicating that the corresponding action for direct activation of the command is a double-click. Menu 801 also includes additional commands 803A that are not directly activatable via a stroke or action.

**[0104]** Referring now to Fig. 12A, there is shown an example of a control region 101 for a text object 102. Text object 102 is a word within a sentence. Control region 101 includes zones 103A through 103J as described above.

**[0105]** Referring now to Fig. 12B, there is shown another example of a control region 101 for text object 102. Again, text object 102 is a word within a sentence, and control region 101 includes zones 103A through 103J as described above. In the example of Fig. 12B, each zone 103 includes a text label to indicate the type of commands available from that zone 103.

**[0106]** Referring now to Fig. 12C, there is shown another example of a control region 101 for text object 102. Again, text object 102 is a word within a sentence, and control region 101 includes zones 103A through 103J as described

above. In the example of Fig. 12C, each zone 103 includes an icon label to indicate the type of commands available from that zone 103.

**[0107]** Referring now to Fig. 12D, there is shown another example of a control region 101 for text object 102. Again, text object 102 is a word within a sentence, and control region 101 includes zones 103A through 103J as described above. In the example of Fig. 12D, a pop-up label 1201 (also known as a ToolTip) appears when cursor 110 hovers over a particular zone 103. The pop-up label 1201 can include text or an icon to indicate the type of commands available from the corresponding zone 103. Pop-up label 1201 can be configured to appear only after cursor 110 has hovered over zone 103 for a predetermined length of time, and can further be configured to disappear when cursor 110 leaves zone 103 and/or after a predetermined period of time. The technique exemplified by Fig. 12D provides a convenient reminder of the functionality for each zone 103 without unduly cluttering the display and without consuming valuable screen real estate. Zones 103 and control region 101 can be made smaller, as there is no need to provide adequate room for text or icon labels within each zone 103.

**[0108]** In the above description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the invention. It will be apparent, however, to one skilled in the art that the invention can be practiced without these specific details. In other instances, structures

and devices are shown in block diagram form in order to avoid obscuring the invention.

**[0109]** Reference in the specification to “one embodiment” or “an embodiment” means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. The appearances of the phrase “in one embodiment” in various places in the specification are not necessarily all referring to the same embodiment.

**[0110]** Some portions of the detailed description are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

**[0111]** It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are

merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the discussion, it is appreciated that throughout the description, discussions utilizing terms such as “processing” or “computing” or “calculating” or “determining” or “displaying” or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system’s registers and memories into other data similarly represented as physical quantities within the computer system’s memories or registers or other such information storage, transmission or display devices.

**[0112]**           The present invention also relates to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus.

**[0113]**           The algorithms and displays presented herein are not inherently related to any particular computer, network of computers, or other apparatus.

Various general-purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct a more specialized apparatus to perform the required method steps. The required structure for a variety of these systems appears from the description. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein.

**[0114]** As will be understood by those familiar with the art, the invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. For example, the particular architectures depicted above are merely exemplary of one implementation of the present invention. The functional elements and method steps described above are provided as illustrative examples of one technique for implementing the invention; one skilled in the art will recognize that many other implementations are possible without departing from the present invention as recited in the claims. Likewise, the particular capitalization or naming of the modules, protocols, features, attributes, or any other aspect is not mandatory or significant, and the mechanisms that implement the invention or its features may have different names or formats. In addition, the present invention may be implemented as a method, process, user interface, computer program product, system, apparatus, or any combination thereof. Accordingly, the disclosure of the present invention is intended

to be illustrative, but not limiting, of the scope of the invention, which is set forth in the following claims.